IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Inventor	:	Jesper Kiehn et al.	Appeal No.
Appln. No.	:	10/815,052	Group Art Unit: 2136
Filed	:	March 31, 2004	Examiner: B. S. Hoffman
For	:	SYSTEM AND METHOD FOR PROVIDING REA MODEL BASED SECURITY	
Docket No.	:	M61.12-0615	

AMENDED BRIEF FOR APPELLANTS

FILED ELECTRONICALLY JANUARY 25, 2007

Sir:

This Amended Brief, in the Appeal from the Office Action dated June 14, 2006 in which claims 1, 3-18, 20-34 and 36-39 were finally rejected, is filed in response to the Notification of Non-Compliant Appeal Brief mailed on December 26, 2006.

Contents

REAL PARTY IN INTEREST

Microsoft Corporation, a corporation organized under the laws of the state of Washington, and having offices at One Microsoft Way, Redmond, Washington 98052, has acquired the entire right, title and interest in and to the invention, the application, and any and all patents to be obtained therefor, as set forth in the Assignment filed with the patent application and recorded on Reel 015174, frame 0984.

NO RELATED APPEALS OR INTERFERENCES

There are no known related appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

STATUS OF THE CLAIMS

I.	Total number of claims in the application.		
		Claims in the application are:	1, 3-18, 20-34 and 36-39
II.	Status of all the claims.		
	A.	Claims cancelled:	2, 19 and 35
	B.	Claims withdrawn but not cancelled:	
	C.	Claims pending:	1, 3-18, 20-34 and 36-39
	D.	Claims allowed:	
	E.	Claims rejected:	1, 3-18, 20-34 and 36-39
	F.	Claims Objected to:	
III.	Claims on appeal		
		The claims on appeal are:	1, 3-18, 20-34 and 36-39

STATUS OF AMENDMENTS

No amendments have been filed after the final rejection. A non-amending response was filed after the final rejection, and has been acted on by the Examiner.

SUMMARY OF CLAIMED SUBJECT MATTER

1. Introduction

The claimed subject matter relates to Resource-Event-Agent (REA) models, and to systems and methods of providing REA model based security.

2. Brief Background

REA is the name of a prescriptive accounting model introduced by William E. McCarthy in 1982. See for example, William E. McCarthy, The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment, *The Accounting Review*, Vol. LVII, No. 3, July 1982. REA is often referred to as a model, a framework, an ontology, an enterprise information system architecture, or by other commonly used names. The fundamental advantage of the REA model is that it provides a prescriptive model for describing a business's processes. Around the fundamental prescriptive model, a whole infrastructure of additions have been added over the years in the form of more specifics on the modeling methodology itself, incorporation of REA in public standards, etc.

While REA allows for the modeling of "ownership" or "involvement", it does not typically address security aspects of business models. Traditional business applications separate security specifics from the domain or business application modeling. Because of this there is very little to discover when it comes to Security configuration or Meta data and the security subsystem is often either missing or implemented in parallel to the application solution.

Prior solutions to security have typically been to let developers set up a list of properties that can or cannot be viewed by roles in the system. This approach is error prone. Further, this approach involves very complex implementations, frequently requiring several days per installation. Sometimes, this approach is implemented in software by the developer coding the solution. This makes it even more difficult to obtain the right security setup as the users (i.e., system administrators, etc) are not able to change the settings and define their own roles/security access.

3. The Claimed Subject Matter

A method of providing REA model based security includes identifying an association between a first object and a second object in an REA model. Then, an association class is created for the association between the first object and the second object. The association class, for example called a Security Policy Association Class, defines security between the first object and the second object.

The association class, defined between the first object and the second object, is an object having properties. The properties of the association class object define the security between the first object and the second object. The step of creating the association class can further comprise creating one or more association class objects having properties, with the properties of the one or more association class objects defining security between a first class of objects of which the first object is a member and a second class of objects of which the second object is a member. The second object can be a securable object, such as a contract or agreement type object, a commitment type object, an event type object, or a resource type object. The first object can be of a particular agent type. A role for a user can be defined by the particular agent type for the first object.

The association class created between the first object and the second object can be created in a security model either separate from the REA model, or as part of the REA model. The security defined between the first object and the second object includes defining permissions and rights of the first object relative to the second object. These permissions and rights can be determined dynamically in a security policy logic module outside of the security model. This is particularly useful for permissions and rights which are transient in nature, for example depending upon the date, time, status of an event, etc.

Independent claim 1 is directed to a method (900) of providing REA model based security.

The method includes the step (905) of identifying an REA defined association (e.g., 515; 615; etc.) of a type which dictates ownership between a first object (e.g., 505; 605, etc.) and a second object (e.g., 510; etc.) in an REA model (e.g., 300; 400; 805; etc.). The method then includes the step (910) of creating an association class object (e.g., 520; 620; etc.) for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object. This method is shown, among other places, in FIG. 9. Other aspects of claimed methods are illustrated in other FIGS., for example in FIGS. 4 through 8-2. This method and related methods are described throughout the application, but particularly between page 21, line 6 and page 31, line 13.

Independent claim 18 is directed to a computer-readable medium having computer-executable instructions for performing steps which are the same as those recited in independent method claim 1. As such, the steps include the step (905) of identifying an REA defined association (e.g., 515; 615; etc.) of a type which dictates ownership between a first object (e.g., 505; 605, etc.) and a second object (e.g., 510; etc.) in an REA model (e.g., 300; 400; 805; etc.). The steps also include the step (910) of creating an association class object (e.g., 520; 620; etc.) for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object. These steps are shown, among other places, in FIG. 9. Other aspects of claimed computer-readable medium steps are illustrated in other FIGS., for example in FIGS. 4 through 8-2. These computer-readable medium steps and related methods are described throughout the application, but particularly between page 21, line 6 and page 31, line 13.

Independent claim 34 is directed to a system for providing security. The system includes an REA model (e.g., 300; 400; 805; etc.) configured to implement a first object (e.g., 505; 605; etc.), a second object (e.g., 510, etc.), and an REA defined association (e.g., 515; 615; etc.) of a type which dictates ownership between the first object and the second object. The system also includes a security model (810) configured to implement an association class object (e.g., 520; 620; etc.) for the REA defined association between the first object and the second object in the REA model, such that properties of the association class object define security between the first object and the second object. This system is shown, among other places, in FIGS. 5-2, 6-2, 8-1 and 8-2. Aspects of this system are described throughout the application, but particularly between page 21, line 6 and page 31, line 13.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1, 3-18, 20-34 and 36-39 are unpatentable under 35 U.S.C. §103(a) over Boozer et al., U.S. Patent Publication 2004/0205355 (hereafter referred to as "Boozer") in view of Tingey, U.S. Patent Publication 2004/0133583.

ARGUMENT

1. CLAIMS 1, 3-18, 20-34 AND 36-39 ARE ALLOWABLE OVER THE CITED BOOZER/TINGEY COMBINATION

Independent claim 1 recites a method of providing Resource-Event-Agent (REA) model based security. The method includes the steps of "identifying an REA defined association of a type which dictates ownership between a first object and a second object in an REA model", and "creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object." Independent claim 18 recites a computer readable medium with the same limitations. Independent claim 34 recites a system for providing security which includes similar limitations. The system includes "a Resource-Event-Agent (REA) model configured to implement a first object, a second object, and an REA defined association of a type which dictates ownership between the first object and the second object." The system of claim 34 also includes "a security model configured to implement an association class object for the REA defined association between the first object and the second object in the REA model, such that properties of the association class object define security between the first object and the second object."

In section 4 of the Final Office Action, claims 1, 3-18, 20-34 and 36-39 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Boozer in view of Tingey. In support of the rejection under 35 U.S.C. § 103(a), the Office Action stated that Boozer teach "a method/system/computer readable medium for providing Resource-Event-Agent (REA) model based security." More specifically, the Office Action stated that Boozer teach the steps of "[i]dentifying an REA defined association of a type which dictates ownership between a first object and a second object," and "[c]reating an association class for the REA defined association between the first object and the second object, the association class defining security between the first object and the second object." The assertion that Boozer teaches these steps is respectfully traversed.

In support of this traversal, it is noted that, also in section 4 of the Office Action, the Examiner again also acknowledged that Boozer does not teach each of these steps. Specifically, the Office Action states that Boozer "does not specifically teach REA models and wherein creating the association class object for the association between the first object and the second object further comprises creating an association class object having properties defining security between the first object and the second object." In fact, however, since Boozer does not teach REA models, REA defined associations of types which dictate ownership, and/or association class objects for the REA defined associations between objects, this reference actually fails to teach or suggest either of the steps of method claim 1. The same is true for the corresponding limitations in computer-readable medium independent claim 18 and system claim 34.

The shortcomings of Boozer in satisfying a *prima facie* conclusion of obviousness against the pending claims are also not overcome by Tingey. The Office Action asserts that Tingey teaches REA models and the limitation of creating association class objects for an association between the first object and the second object, with the association class object having properties defining security between the first object and the second object. Specifically, the Office Action references paragraph 0066 of Tingey as providing such a teaching. These assertions regarding the disclosure of Tingey are respectfully traversed as well.

Tingey teach a record-extensible event accounting structure or approach which is compatible with the resource, event and agent orientation of the REA model. See for example, Tingey at paragraphs 0009, 0059 and 0060. As such, Tingey makes general references to REA models and some aspects of REA model structure. However, like Boozer, Tingey does not teach the step of "creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object," which is recited in independent claims 1 and 18. Nor does Tingey teach the similar limitation in independent system claim 34 of "a security model configured to implement an association class object for the REA defined association between the first object and the second object in the REA model, such that properties of the association class object define security between the first object and the second object." In fact, the Tingey publication does not show, discuss, or make any reference to association class objects for REA defined associations between a first object and a second object. Without teaching the association class object recited in the rejected claims, it is not possible for Tingey to teach that the association class object has properties defining security between the first and second objects, as is also specifically required in each of the rejected claims.

In paragraph 0066 of Tingey, which was cited by the Office Action as teaching association class objects and the definition of security using association class objects, no such teaching is actually provided. Paragraph 0066 of Tingey states that:

> Security and stability of data in the proposed architecture are factors in the selection of standardized event summary and detail records. Of course, a record-extensible structure, such as is described herein, is possible only through use of classification and hierarchy establishing tools and concepts along with relational models. Use of both kinds of models are critical to successful implementation of a functional security model. By definition, security itself is a hierarchical phenomenon, namely that rights are granted to individuals and organizations based on some form of classification. Thus, an approach based on hierarchic as well as relational structures is viable to the degree that such tree-based classification systems are available to secure and to organize the data. As an example of how such a record-extensible environment functions, three composite "Big E" events are outlined.

While Tingey does briefly mention the general concept of "security", this reference does not teach or suggest that security between a first object and a second object is defined in an association class object created for an REA defined association between the first and second object. Instead, Tingey only state that security is based upon granting rights to individuals and organizations based some form of classification using tree-based classification systems.

As is well established, the Examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness. If the Examiner does not produce a *prima facie* case, the Applicant is under no obligation to submit evidence of nonobviousness." See MPEP § 2142. "To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest ***all*** the claim limitations." (Emphasis added). See MPEP § 2142.

It has been shown that neither of Boozer or Tingey teach or suggest the limitation found in independent claims 1 and 18 of "creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object." Using the same analysis, it has been shown that neither of Boozer or Tingey teach or suggest the claim limitation found in independent claim 34 of "a security model configured to implement an association class object for the REA defined association between the first object and the second object in the REA model, such that properties of the association class object define security between the first object and the second object." Since the combination of Boozer and Tingey do not teach or suggest all of the claim limitations, a *prima facie* case of obviousness has not been established for any of the independent or dependent claims. Consequently it is respectfully requested that the Board reverse the Examiner's rejection of independent claims 1, 18 and 34, as well as the rejection of the dependent claims depending therefrom.

In addition to being in allowable form based upon their dependence from independent claims 1 and 18, dependent claims 11, 12, 27 and 28 contain additional limitations which are neither taught nor suggested by the cited Boozer/Tingey combination. Dependent claims 11 and 27 recite the claim limitation "wherein identifying the REA defined association of the type which dictates ownership between the first object and the second object further comprises identifying an REA defined control type association between the first object and the second object". Dependent claims 12 and 28 recite the limitation "wherein identifying the REA defined association of the type which dictates ownership between the first object and the second object further comprises identifying an REA defined custody type association between the first object and the second object".

In support of the rejection of claims 11, 12, 27 and 28, the Office Action states that the combination of Boozer and Tingey teaches "wherein identifying the REA defined association of the type which dictates ownership between the first object and the second object further comprises identifying an REA defined [control type/custody type] association between the first object and the second object (see page 1, paragraph 0016 and page 3, paragraph 0033 of Boozer et al., control meaning 'ownership' and custody meaning 'template')". These interpretations of the Boozer reference are respectfully traversed. Boozer provides no such teaching or suggestion. For example, at page 1, paragraph 0016 cited in the Office Action, Boozer states:

> FIG. 2 depicts a data access security system 60 for accessing information stored within a data storage unit or units. The stored information is retrieved through resource objects (70, 74, 76, 78) which are interconnected through a complex set of relationships or associations 72. Associations 72 define conceptual relationships between an instance of one class and an instance of another class. For example, objects may be associated to higher level container objects, such as a column object having an association to the table within which it is found as well as to other objects. The table object may itself have multiple associations, such as to libraries and trees within which it is located. In this respect, an object 70 may have associations 72 to multiple objects (74, 76, 78).

At page 3, paragraph 0033 cited in the Office Action, Boozer states:

> If the routine at decision process 310 returns a NOTFOUND result, then process 316 locates the default access control template for the repository where the resource object resides. Decision process 318 examines whether a default access control template was found. If not, then at 320 the result is cached as a GRANT permission and a true value is returned. If a default access control template was found as determined by decision process 318, then decision process 322 invokes the CheckACT routine (which is described in FIG. 13). If the result from the routine is a DENIED result, then the result is cached and a false value is returned at 324. However, if a GRANTED value is returned, then at 326 the result is cached and a true value and NO CONDITION result are returned. If the routine returns a NOTFOUND, then at 328 the result is cached as DENIED and returns a false value.

Contrary to the assertions made in the Office Action, these cited portions of Boozer do not teach or suggest REA defined control type associations or custody type associations between first and second objects. Further, the Examiner's statement that "control meaning 'ownership' and custody meaning 'template'" is not supported by the teachings of the Boozer reference. Boozer does not teach REA defined "ownership". Also, there is no teaching that supports an analysis of Boozer to interpret custody as meaning "template". Lacking a teaching or suggestion of the additional claim limitations found in dependent claims 11, 12, 27 and 28, it is respectfully requested that the rejection of these claims be reversed for these additional reasons.

In section 5 of the Final Office Action, the Examiner provided responses to previously submitted arguments. These responses are addressed in this Appeal Brief to further clarify the issues before the Board. First, the Examiner stated that:

> Applicant argues:
> a. Boozer does not teach REA models, particularly 'creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object' (page 9, second paragraph).

In response, the Examiner stated that:

> Regarding argument (a), examiner disagrees with applicant. Boozer shows associations between objects to determine security between objects. In figure 2,

> the first object and the second object are in a relationship with each other, as imposed by the containment boundary. The containment boundary establishes security rules for the two objects, which would be an association class object. As for Boozer not showing REA model, *ipsissimis verbis* states that the elements must be arrange (sic) as required by the claims, but the terminology or wording is not required. See In re Bond, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

The Examiner's analysis is respectfully traversed. First, the fact that the first object and the second object of Figure 2 of Boozer are in a relationship with each other does not support the Examiner's statement "as imposed by the containment boundary." The fact that Boozer teaches associations between objects is not disputed. However, Boozer does not state that the association between the first and second objects is imposed by the containment boundary. Instead, Boozer teach that in forming the containment boundary, the security rules specify which object associations are involved. For example, on page 1, paragraph [0018], Boozer states:

> Object security rules 66 help form the containment boundary 68 by specifying what object associations 72 are involved in constructing the containment boundary 68. Because, an object 70 may have multiple associated objects (74, 76, 78), object access security rules 66 specify which object associations 72 are to be used in constructing the containment boundary 68.

Second, the Examiner's statement that "[t]he containment boundary establishes security rules for the two objects, which would be an association class object" (emphasis added) is without support in Boozer. Specifically, the second part of that statement does not follow from the first. As noted, Boozer has stated that the security rules help form the containment boundary. More important, however, is the fact that there is nothing about a containment boundary establishing security rules (or vice versa in the above Boozer quote where the security rules construct the containment boundary) for two objects which would necessitate or support the conclusion "which would be an association class object." Boozer does not teach or suggest creating an association class object for an association between the first object and the second object, with the association class object having properties defining security between the first object and the second object, neither for an REA defined association or for a non-REA defined association.

Third, the Examiner's reliance on the statement, that "[a]s for Boozer not showing REA model, *ipsissimis verbis* states that the elements must be arrange (sic) as required by the claims, but the terminology or wording is not required", is traversed. This principle is not applicable in the present instance. In the case at hand, even ignoring for the moment the REA model limitation of the claim, it has been demonstrated that the elements taught by Boozer are not arranged as required by the claims. Boozer does not teach the creation of an association class object for an association between first and second objects, with the association class object having properties defining security between the first object and the second object. Further, this statement by the Examiner falsely presumes that claim limitations such as "REA model" and "REA defined association" are merely terminology. In fact, as has been pointed out, this is not the case and these express claim limitations cannot be simply ignored.

Also in section 5 of the Final Office Action, the Examiner stated that:

> Applicant argues:
>
> a. . . .
>
> b. Tingey does not teach REA security, namely 'creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object' (page 9, last paragraph through page 10, first paragraph).

In response, the Examiner stated that:

> Regarding argument (b), examiner disagrees with applicant. Boozer was stated for teaching all of the limitations of claim 1, except that there was no stated REA model and the association class object has properties for defining security. Tingey discloses security of data (or objects) with properties.

The Examiner's analysis is again respectfully traversed. First, in the Office Action, Tingey was cited for teaching more than security of data or objects with properties. The rejection in section 4 (page 3) of this Final Office Action states:

> Tingey teaches REA models (fig. 1), and wherein creating the association class object for the association between the first object and the second object further comprises creating an association class object having properties defining security between the first object and the second object (paragraph 0066).

As has been shown in the previously filed Response, like Boozer, Tingey does not teach the step of "creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object." Neither reference teaches this claim limitation. Nor does either reference teach the similar limitation in other claims of "a security model configured to implement an association class object for the REA defined association between the first object and the second object in the REA model, such that properties of the association class object define security between the first object and the second object." As has been demonstrated, like Boozer, the Tingey publication does not show, discuss, or make any reference to association class objects for REA defined associations between a first object and a second object. Without one of these two cited references teaching the association class object recited in the rejected claims, it is not possible for either reference to teach that the association class object has properties defining security between the first and second objects, as is also specifically required in each of the rejected claims. Lacking a teaching of these claim limitations in either reference, a *prima facie* case of obviousness has not been made, and the claims are believed to be in allowable condition. Reversal of all rejections made in the Final Office Action is therefore respectfully requested.

2. CONCLUSION : CLAIMS 1, 3-18, 20-34 AND 36-39 SHOULD BE ALLOWED

In conclusion, the Appellants respectfully submit that claims 1, 3-18, 20-34 and 36-39 are allowable over the cited references for at least the reasons laid out above. Thus, the Appellants respectfully request that the Board reverse the rejections of claims 1, 3-18, 20-34 and 36-39 and find the claims in condition for allowance. The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /John D. Veldhuis-Kroeze/
John D. Veldhuis-Kroeze, Reg. No. 38,354
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3244
Phone: (612) 334-3222
Fax: (612) 334-3312

JVK/jme

Appendix A: Claims On Appeal

Claims as they currently stand:

1. A method of providing Resource-Event-Agent (REA) model based security, the method comprising:

identifying an REA defined association of a type which dictates ownership between a first object and a second object in an REA model;

creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object.

2. (canceled)

3. The method of claim 1, wherein creating the association class object further comprises creating one or more association class objects having properties, the properties of the one or more association class objects defining security between a first class of objects of which the first object is a member and a second class of objects of which the second object is a member.

4. The method of claim 1, wherein the second object is a securable object.

5. The method of claim 4, wherein the first object is of a particular agent type, and wherein a role for a user is defined by the particular agent type for the first object.

6. The method of claim 5, wherein the second object is a contract or agreement type object.

7. The method of claim 5, wherein the second object is a commitment type object.

8. The method of claim 5, wherein the second object is an event type object.

9. The method of claim 5, wherein the second object is a resource type object.

10. The method of claim 5, wherein the second object is an agent type object.

11. The method of claim 5, wherein identifying the REA defined association of the type which dictates ownership between the first object and the second object further comprises identifying an REA defined control type association between the first object and the second object.

12. The method of claim 5, wherein identifying the REA defined association of the type which dictates ownership between the first object and the second object further comprises identifying an REA defined custody type association between the first object and the second object.

13. The method of claim 5, wherein creating the association class object for the REA defined association between the first object and the second object further comprises creating the association class object in a security model.

14. The method of claim 13, wherein creating the association class object in the security model further comprises creating the association class object in the security model separate from the REA model.

15. The method of claim 13, wherein creating the association class object in the security model further comprises creating the association class object in the security model as part of the REA model.

16. The method of claim 13, wherein defining security between the first object and the second object further comprises defining permissions and rights of the first object relative to the second object.

17. The method of claim 16, wherein defining permissions and rights of the first object relative to the second object further comprises dynamically determining the permissions and rights in a security policy logic module outside of the security model.

18. A computer readable medium having computer-executable instructions for performing steps of a method of providing Resource-Event-Agent (REA) model based security, the steps comprising:

identifying an REA defined association of a type which dictates ownership between a first object and a second object in an REA model;

creating an association class object for the REA defined association between the first object and the second object, the association class object having properties defining security between the first object and the second object.

19. (canceled)

20. The computer readable medium of claim 18, wherein creating the association class object further comprises creating one or more association class objects having properties, the properties of the one or more association class objects defining security between a first class of objects of which the first object is a member and a second class of objects of which the second object is a member.

21. The computer readable medium of claim 18, wherein the first object is of a particular agent type, and wherein a role for a user is defined by the particular agent type for the first object.

22. The computer readable medium of claim 21, wherein the second object is a contract or agreement type object.

23. The computer readable medium of claim 21, wherein the second object is a commitment type object.

24. The computer readable medium of claim 21, wherein the second object is an event type object.

25. The computer readable medium of claim 21, wherein the second object is a resource type object.

26. The computer readable medium of claim 21, wherein the second object is an agent type object.

27. The computer readable medium of claim 18, wherein identifying the REA defined association of the type which dictates ownership between the first object and the second object further comprises identifying an REA defined control type association between the first object and the second object.

28. The computer readable medium of claim 18, wherein identifying the REA defined association of the type which dictates ownership between the first object and the second object further comprises identifying an REA defined custody type association between the first object and the second object.

29. The computer readable medium of claim 18, wherein creating the association class object for the REA defined association between the first object and the second object further comprises creating the association class object in a security model.

30. The computer readable medium of claim 29, wherein creating the association class object in the security model further comprises creating the association class object in the security model separate from the REA model.

31. The computer readable medium of claim 29, wherein creating the association class object in the security model further comprises creating the association class object in the security model as part of the REA model.

32. The computer readable medium of claim 29, wherein defining security between the first object and the second object further comprises defining permissions and rights of the first object relative to the second object.

33. The computer readable medium of claim 32, wherein defining permissions and rights of the first object relative to the second object further comprises dynamically determining the permissions and rights in a security policy logic module outside of the security model.

34. A system for providing security, the system comprising:

a Resource-Event-Agent (REA) model configured to implement a first object, a second object, and an REA defined association of a type which dictates ownership between the first object and the second object;

a security model configured to implement an association class object for the REA defined association between the first object and the second object in the REA model, such that properties of the association class object define security between the first object and the second object.

35. (canceled)

36. The system of claim 34, wherein the association class object further comprises one or more association class objects having properties, the properties of the one or more association class objects defining security between a first class of objects of which the first object is a member and a second class of objects of which the second object is a member.

37. The system of claim 34, wherein the security model is separate from the REA model.

38. The system of claim 34, wherein the security model is part of the REA model.

39. The system of claim 34, and further comprising a security policy logic module coupled to the security model and configured to dynamically determine permissions and rights of the first object relative to the second object.

Appendix B: Evidence Appendix

There is no known evidence submitted pursuant to 37 CFR §§ 1.130, 1.131 or 1.132 or other evidence entered by the Examiner.

Appendix C: Related Proceedings Appendix

There are no known related appeals or interferences regarding the present appeal.